Travaux Pratiques ML No.3

Cours d'Informatique de Deuxième Année

—Deug SSM M4—

Aujourd'hui...

On continue avec les listes et le pattern-matching.

Pattern Matching et listes

▸ **Exercice 2**. Ecrire une fonction de type `int list -> int list` qui prend la liste $[x_1; \ldots; x_n]$ et qui retourne $[x_1 + 1; \ldots; x_n + 1]$ en utilisant la fonction `map`.

. . . . . .

▸ **Exercice 3**. Ecrire une fonction qui à la liste $[x_1; \ldots; x_n]$ associe la liste [fib x_1; ...; fib x_n] où `fib` est la fonction calculant les éléments de la suite de Fibonacci.

. . . . . .

▸ **Exercice 4**. Ecrire une fonction[1] `g` définie par itération sur les listes à partir de `a` et de `f` et donnée par les relations suivantes:

$$\begin{cases} g(\ []\ ) = a \\ g(\ x :: l\ ) = f\ x\ (g\ l) \end{cases}$$

. . . . . .

. . . . . .

Les exercices suivants sont à faire à l'aide des fonctionnelles sur les listes `list_it` et `map`.

▸ **Exercice 5**. Ecrire une fonction qui à $[x_1; \ldots; x_n]$ associe $\sum_{i=1}^{n} x_i$.

. . . . . .

▸ **Exercice 6**. Ecrire une fonction qui renvoie la conjonction de tous les éléments d'une liste[2].

. . . . . .

▸ **Exercice 7**. Ecrire une fonction qui calcule la longueur d'une liste.

. . . . . .

[1] dont on donnera le type avant de l'écrire.

[2] Quel est le type des éléments de la liste ?

▸ **Exercice 8**. Réécrire la fonction de concaténation de deux listes.

. . . . . .

▸ **Exercice 9**. Ecrire une fonction `Rev` qui renverse une liste.

. . . . . .

▸ **Exercice 10**. Ecrire la fonction `Flat` qui à la liste $[[1;2;4];[4;5]]$ associe la liste $[1;2;4;4;5]$.

. . . . . .

▸ **Exercice 11**. Ecrire la fonctionnelle `map` avec `list_it`.

. . . . . .

On définit un ensemble comme une liste sans répétition.

▸ **Exercice 12**. Ecrire une fonction `Vide` qui teste si un ensemble est vide.

. . . . . .

▸ **Exercice 13**. Ecrire une fonction `Belong` qui teste l'appartenance d'un élément à un ensemble.

. . . . . .

▸ **Exercice 14**. Ecrire une fonction `Insert` qui ajoute un élément dans un ensemble.

. . . . . .

▸ **Exercice 15**. Ecrire une fonction `Delete` qui supprime un élément d'un ensemble.

. . . . . .

▸ **Exercice 16**. Ecrire une fonction `List2ens` qui transforme une liste en un ensemble.

. . . . . .

▸ **Exercice 17**. Ecrire une fonction `Union` qui fait l'union de deux ensembles.

. . . . . .

▸ **Exercice 18**. Ecrire une fonction `Inter` qui fait l'intersection de deux ensembles.

. . . . . .

▸ **Exercice 19**. Ecrire une fonction `Prod_cart` qui fait le produit cartésien de deux ensembles.

. . . . . .

Exercice supplémentaire

▸ **Exercice 20**. Ecrire une fonction qui prend en argument une liste `l` assimilée à un ensemble et qui retourne l'ensemble des parties de `l`.